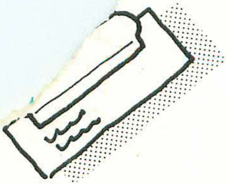
解る

CASI



PV-7 [ベーシックガイド]

MSX



この本の使い方

MSX BASICには、たくさんの働きがあります。

この本では、その中からゲーム作りに特に便利なものをピックアップして紹介していきます。その働きは、本の終わり(P42)に一覧表にしてありますので参考にしてね。

資料編のさくいんは、"あれは、本のどのあたりに書いてあったかな?"というときに助けてくれる便利なページです。おおいに利用してください。

みなさん、またお会いしましたね。バーネットおばさんです。

PV-7の使い方に少しなれてきたところかな？

それじゃ、こんどは、いよいよBASICを覚えてゲームを作ってみましょう。

ひとつのゲームを少しずつ作っていくうちにとっても面白いゲームのプログラムができちゃうの。

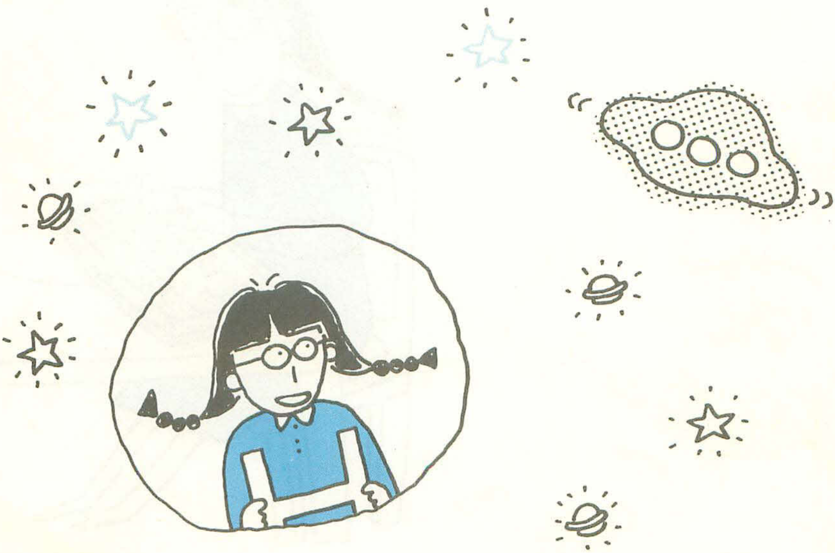
今度もおばさんといっしょに遊びましょう。遊んでるうちに、いろんなことも覚えられちゃうからね。

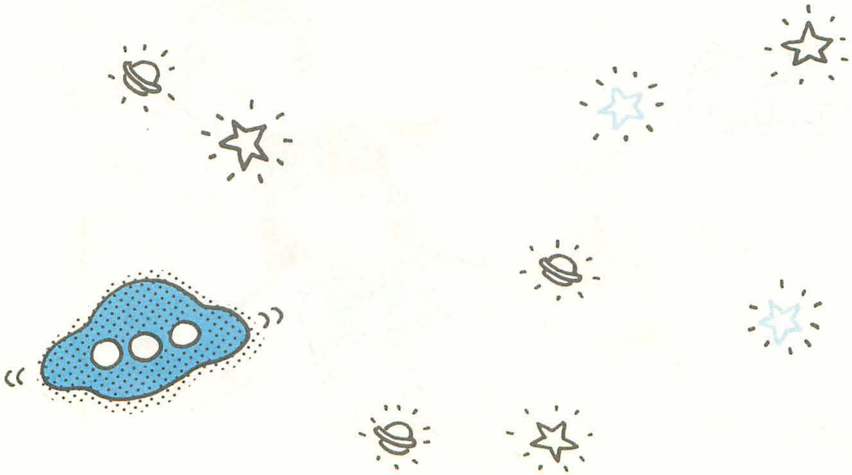
さあ、キミたちもPV-7でゲームを作っちゃおう！

それでは、電源をONにするところからスタートしましょうね。



もくじ





さあ始めましょう

●**電源 ON で BASIC の起動**

PV-7とテレビ(またはAVテレビ)はきちんとつながってるかな？つながっていたら、テレビのスイッチとPV-7のスイッチをONにします。

そうすると右のような画面が現われてBASICのはじまり、はじまり。

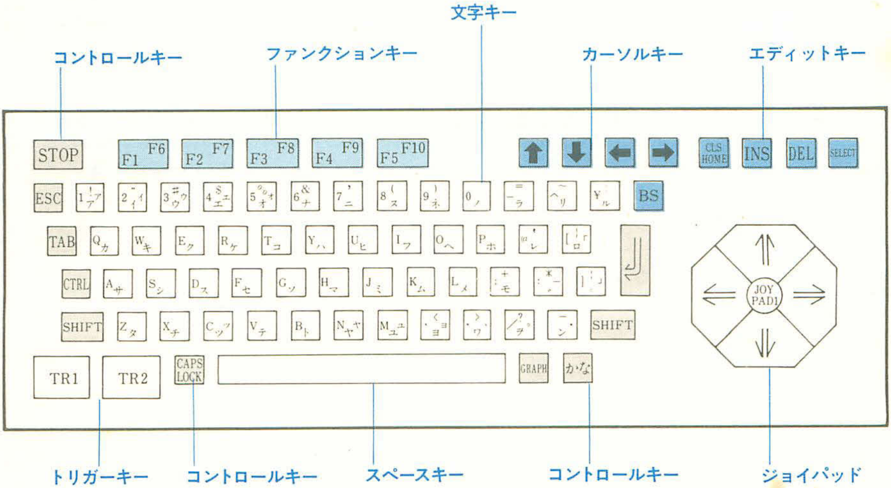
もし、うまくこの画面が出てこなかったら、取扱説明書の〝ステップ2しっかりつなごう〃のページをみてね。





キーボードになれよう

PV-7 のキーボードをみてみましょうね。これが友だちになる第2歩め。キーがいっぱい並んでますね。かなにアルファベットにいろいろな記号。その他のいろいろなキーにも、それぞれ便利な働きがありますからね。どんどん覚えて、使っちゃいましょう。



じつは、キーボードは１つのキーで何種類かの文字を表示することができます。その秘密は、SHIFT と CAPS LOCK と かな と GRAPH にあるんでしたね。

この４つのキーを押すと PV-7では、アルファベットの大文字と小文字、ひらがな、カタカナ、グラフィック記号などが全部表示できるのです。4つのキーのうち１番活躍するのが SHIFT です。

SHIFT を押しながらキーを押すとキーのおでこに書いてある記号がでてきます。また、アルファベットのキーといっしょに押せば大文字がでてきますよ。

2つのキーをいっしょに押すときは↗の記号を使いますよ。たとえば、

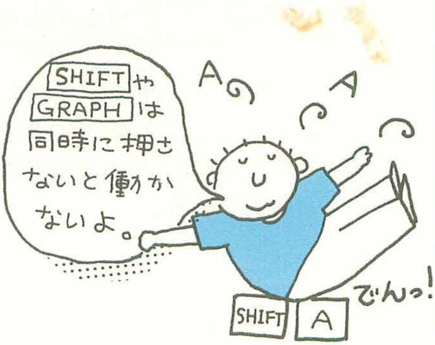
SHIFT ↗ A とかね。



●コントロールキーの秘密

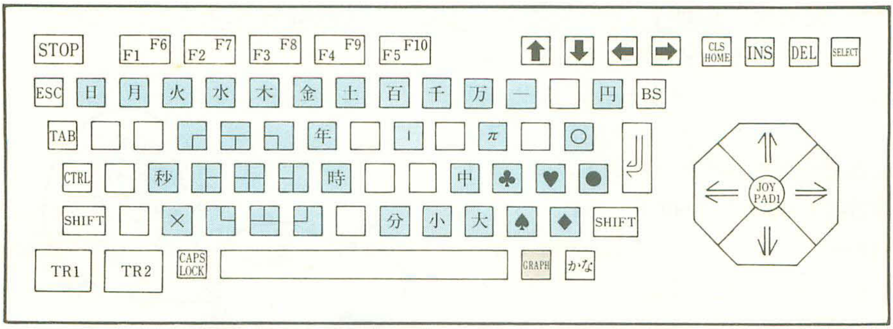
CAPS LOCK 、かな 、GRAPH 、SHIFT のコントロールキーを押すことによって下の表のように文字のキーが切りかわります。

CAPS LOCK と かな は１回押すとずっとその働きをします。もう１度押すともどります。SHIFT と GRAPH はそのつど押しながら使います。



キー	文字の種類と位置	SHIFT を押しながらのとき
何も押してない	アルファベット小文字・キーの左	アルファベット大文字・キーの上
CAPS LOCK	アルファベット大文字・キーの左	アルファベット小文字・キーの上
かな	ひらがな・キーの下	キーの右
CAPS LOCK かな	カタカナ・キーの下	キーの右
GRAPH	グラフィックパターン	

キーには書いていないけど、GRAPH を押しながらキーを押すと下のようなグラフィックパターンを表示します。たとえば、GRAPH ⤴ : なら♥とかね。



●その他のコントロールキー

コントロールキーには他にも ↵(リターン) や TAB(タブ) や CTRL(コントロール) があります。

↵：キーボードからいれた命令やデータを PV-7 に伝える働きをします。命令などをいれた後には、必ず押しましょうね。

TAB：画面には横に 8 文字ずつカーソルをジャンプさせるタブがセットされています。TAB を押すとカーソルのところから右の 1 番近いタブのところまでカーソルがジャンプ。そのときに文字があるところをカーソルが通り過ぎると文字は消えてしまいますよ。

CTRL：CTRL を押しながらキーを押すと、こんな働きをします。

キー	CTRL といっしょに押したときの働き
B	カーソルを 1 つ前の単語の最初に動かす。
C	入力待ち状態を中断する。
E	カーソルのある行のカーソルの後の文字などを消す。
F	カーソルを 1 つ後の単語の最初に動かす。
G	〝ピッ〟という音を出す。
H	カーソルのすぐ前の文字を消す。（ BS と同じ）
I	TAB と同じ働き。
J	カーソルを次の行に動かす。
K	カーソルを画面の左上の端に動かす。（ HOME(ホーム) と同じ）
L	画面の文字を消してカーソルを画面の左上の端に動かす。（ CLS(クリアスクリーン) と同じ）
M	↵ と同じ働き
N	カーソルを行の最後に動かす。
R	インサートモードにする。（ INS(インサート) と同じ）
U	カーソルのある行を消す。
¥	→ と同じ働き。
]	← と同じ働き。
^	↑ と同じ働き。
—（アンダーライン）	↓ と同じ働き。

●まちがえたらエディットキー

エディットキーはプログラムの修正をするときなどに活躍します。

キー	働き
↑ ↓ ← →	カーソルを上下左右に１文字分ずつ動かす。
INS	文字と文字の間に文字を入れる。文字を入れたいところにカーソルを動かし、 INS を押してカーソルが小さくなったらインサートモード。これでキーを押せばカーソルの前に文字が入っていく。 インサートモードをやめるにはもう１度 INS を押すか、↑ ↓ ← → を押す。
デリート DEL	カーソルのあるところの文字を消す。
HOME	カーソルを画面の左上の端に動かす。
CLS SHIFT ↱ HOME	画面の文字を消して、カーソルを画面の左上の端に動かす。
BS	カーソルの１文字前に戻って文字を消す。
ストップ STOP	プログラムの実行やリストの表示を途中で止める。 止まっているときに他のキーを押しても何もうちこめない。 もう１度 STOP を押すとまた始まる。
CTRL ↱ STOP	プログラムの実行やリストの表示を途中で止める。 プログラムの実行を続けたいときは C O N T ↲ で止めたところから始まる。リストのときは、つづきはでてこない。

ジョイパッドにも矢印はついているけど、カーソルキーとはちがって、BASICのときは、つかいませんよ。

●ファンクションキー

ファンクションキー1つを押すだけで、下のようにキーを押したのと同じ働きになります。これはらくですね。

キー	文字キーでいれたとき	キー	文字キーでいれたとき
F1	C O L O R スペース	F6	C O L O R スペース 1 5 , 4 , 7 RETURN
F2	A U T O スペース	F7	C L O A D "
F3	G O T O スペース	F8	C O N T RETURN
F4	L I S T スペース	F9	L I S T . RETURN
F5	R U N RETURN	F10	C L S R U N RETURN



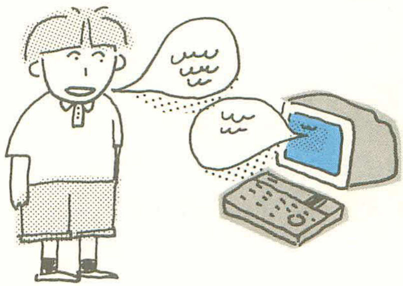
さあ、次はワクワクドキドキのBASIC。どんどんキーを押してキーボードになれながらBASICも覚えちゃいましょう。

MSX BASICってなあに?

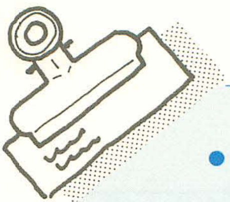
キミたちが言葉のちがう国の人と話すとき、相手の人の国の言葉を使ったり、相手の人が日本語を使ってなんとか話しが通じるように努力しますよね。PV-7と話すときも、キミたちがPV-7にわかる言葉で話さないと通じません。このPV-7にわかる言葉というのがBASICなんです。

ところが、今までは同じBASICでもコンピュータを作っている会社がちがうと細かい部分が違っていて、ちがうコンピュータだと同じプログラムは使えなかったの。

そこで、どのコンピュータのBASICでもみんな同じに使えるようにしようというのがMSX BASIC。MSXならちがう会社のコンピュータで作ったプログラムやゲームが平気で使えちゃうんです。



だから、キミたちの友だちにMSXパソコンを持ってる子がいたら、おたがいに自分たちの作ったゲームプログラムのとりかえっこができるというわけです。すごいでしょう。



●プログラムって何?

コンピュータ・ゲームでキミたちも遊ぶでしょ?UFOが飛んだり、ミサイルを発射したり、ストーリーがあるよね。そのストーリーをつくってあるのがプログラムなの。コンピュータにしてほしい仕事を行番号という番号をつけてならべたものなんです。プログラムをいれたら"RUN"という命令で動きだすんです。電源を切ったり"NEW"という命令でプログラムを消さない限り何回でも遊んだり、ためしてみたりすることができるんですよ。

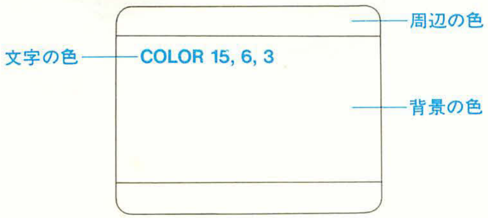
なにができるかな

これからプログラムを作るにはどうしたらいいかってことをお話するわけだけど、そのまえにちょっとウォーミングアップ。キーを押してみましょう。さあ、画面にはなにがでるかな？

●画面の色を変えてみよう

カラー COLOR	画面の色を決める
書き方	COLOR　文字の色，背景の色，周辺の色
使い方	COLOR 15, 6, 3

キーボードから C O L O R スペース 1 5 , 6 , 3 ↵ といれてみましょう。画面の色が赤に変わります。色は右の表のように番号で選びます。画面や文字の色をプログラムの中で変えるときにも使いますよ。



コード	色	コード	色
0	透明	8	赤
1	黒	9	明るい赤
2	緑	10	黄
3	明るい緑	11	明るい黄
4	暗い青	12	暗い緑
5	明るい青	13	紫
6	暗い赤	14	灰
7	水色	15	白

もし、色を変えたときに文字が見えなくなったりしたら、SHIFT ↗ F1 を押せばもとの色にもどります。

●画面をきれいに

クリアスクリーン CLS	画面の文字や数字を消す
書き方	CLS
使い方	

C L S ↵ と入力すると今まで画面にでていた文字や数字が全部消えます。CLS は SHIFT ↗ HOME でも同じ。プログラムの中で画面をきれいにしたいとき活躍する命令です。



●いろんなところにいろいろ書こうPRINT, LOCATE

PRINT と LOCATE を組みあわせると画面の好きな場所に好きな文字などをだせます。
[1][0][スペース][L][O][C][A][T][E][5][,][5][↵][2][0][スペース][P][R][I][N][T][〝][オ][ハ][゛][サ][ン][〞][↵]といれましょう。

すると右のように〝オバサン〞という文字をだしましたね。こんなふうに〝(ダブルクォーテイション)で囲んだものを画面の決められた位置にだすのです。

さて、PRINT を使うときには次のような約束があります。

プリント PRINT	画面に文字や数字や記号をだす
書き方	PRINT 〝だしたいもの〞
使い方	PRINT 〝オバサン〞

ロケート LOCATE	画面にだす文字や数字の位置を決める
書き方	LOCATE ヨコの位置、タテの位置
使い方	LOCATE 1Ø0, 5Ø

```
1Ø LOCATE 4, 5
2Ø PRINT 〝オバサン〞

    オバサン
    ↑
OK  ここが5行目の
      4文字目
□
```

(この文字をだす)

PRINT だけなら	;(セミコロン)をつけると	,(コンマ)で区切ると
PRINT OK □ 改行する	1Ø PRINT 〝オバサン〞; 2Ø PRINT 〝36サイ〞 RUN オバサン36サイ OK □ 続けてだす	PRINT 〝PV-7〞, 〝MSX〞 PV-7　　　　MSX OK □ 15文字目ごとにだす

LOCATE を使うときには、次に説明する画面の文字数と行数の中で位置を決めます。

●画面の切りかえ

画面にはテキスト画面とグラフィック画面の2つがあります。電源を ON にしてからキミたちが使っているこの画面はテキスト画面といいます。

テキスト画面	グラフィック画面
文字や数字を書く	絵を描く

スクリーン SCREEN	画面を切りかえる
書き方	SCREEN 画面,スプライトの大きさ
使い方	SCREEN Ø, 2

画面:
- Ø:ヨコ4Ø×タテ24文字のテキスト画面
- 1:ヨコ32×タテ24文字のテキスト画面
- 2:ヨコ256×タテ192ドットの高解像度グラフィック画面
- 3:ヨコ64×タテ48ブロックのマルチカラー画面

スプライトの大きさ:
- Ø:8×8ドット
- 1:8×8ドットで2倍にしてだす
- 2:16×16ドット
- 3:16×16ドットで2倍にしてだす

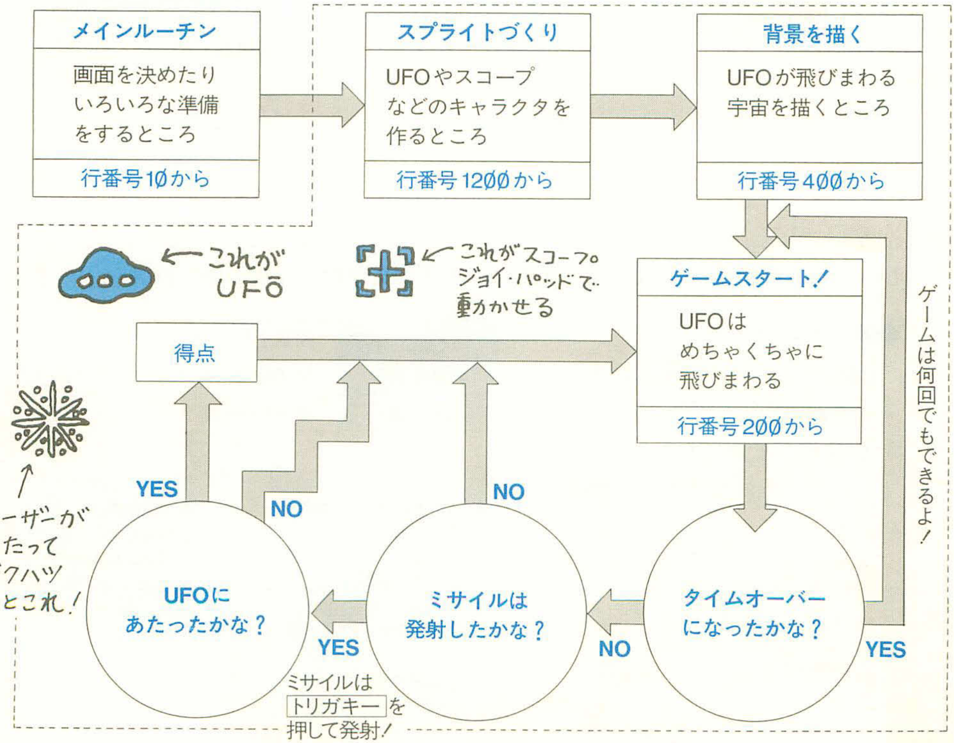
ゲームをつくってみよう

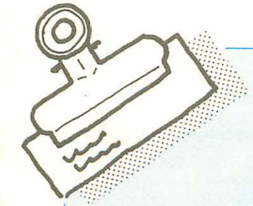
ゲームをつくろう① こんなゲーム

さて、いよいよゲームプログラムづくりを始めましょう。これから作るのは、なづけて『UFO げきついスコープゲーム』。キミたちがいるのは宇宙船スターシップ号のそうじゅう席。レーザー砲のスコープをのぞいて、飛んできた UFO にスコープをあわせて、レーザー発射。UFO をやっつければ、はい得点。自分で作ったプログラムで遊べるなんて最高でしょ？

それじゃ、このゲームがどういうふうにできるか、ちょっと見てみましょう。

プログラムは、メインルーチンというプログラムの流れや、最初にでてくるものを決めておいたりするところと、サブルーチンという、いろいろな仕事をするところでできているのですよ。





プログラムをいれよう

　これからゲームを作るプログラムがでてくるから、リストを見ながらこんなふうにいれていってね。ゆっくり、まちがえないようにキーを押していきましょう。

1 まず、[N][E][W][↵]

```
NEW
OK
□
```
前のプログラムを消す

2 次に [A][U][T][O][1][2][0][0][↵]

```
AUTO 1200
1200 □
```
↑これが行番号

AUTO 200ならば200からでるよ

3 プログラムリストを見ながら [1][0]['][*][*][*][*][*][ス][プ][ラ] ……と押していく

4 プログラムの行の終わりにきたら [↵] をおす

```
1200'***スプライト***[↵]
1210 □
```
└[↵]を押すと次の行番号がでる

5 同じように次の行もいれていく

```
200 '*****ウゴ カス***
210 ON SPRITE GOSU
220 SPRITE OFF STR
240 X1=INT(RND(1)*
*30)*5+25
250 FOR K1=1 TO 30
260 SPRITE OFF STR
270 A=STICK(1)
280 IF (A=2 OR A=3
```
こういうところがプログラムの行の終わり

6 最後までできたら [CTRL] ↗ [STOP] でできあがり

```
1420 DATA 20……
1430
OK
□
```

もしも、押しまちがえたりしたらこんなふうに直せばいいの。

[↵] を押す前にまちがいに気づいたら

1 [←] でまちがえたところにカーソルをもどす

```
40 SCRE[R]N 2,2
```

2 正しいキーを押す

```
40 SCREE[N] 2,2
```

3 [↵] で次の行に進む

```
40 SCREEN 2,2
50 □
```

RUNしてエラーメッセージがでたら

1 [R][U][N][↵]

```
RUN
Syntax error in 40
OK
□
```
↑行番号40がちがうという意味

2 [L][I][S][T][4][0][↵]

```
LIST 40
40 SCREEM 2,2
OK
□
```
↖指定の行番号をだす

3 [↑][→] でカーソルをまちがえているところに動かす

```
40 SCREE[M] 2,2
OK
```

4 正しいキーを押して直す

```
40 SCREEN□2,2
OK
```

5 [↵] を押す

```
40 SCREEN 2,2
OK
□
```
[↵] を忘れると直らないので気をつけよう



あ! UFOだ

ゲームをつくろう② キャラクタをつくる

まずは、ゲームの中のキャラクタを作ってみましょう。プログラムリストの行番号はあとから追加になったりするのでこのとおりにいれていってね。

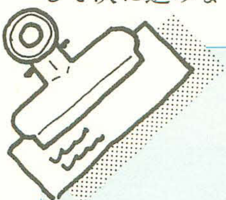
```
10 '*****メイン ルーチン*****
40 SCREEN 2,2
50 COLOR 15,1,1 :CLS
70 GOSUB 1200 'スプライト
```

```
1200 '*****スプライト*******
1210 RESTORE 1280
1220 FOR P=0 TO 2:B$=""
1230 FOR I=1 TO 32
1240 READ A$
1250 B$=B$+CHR$(VAL("&H"+A$)):NEXT I
1260 SPRITE$(P)=B$:NEXT P
1270 RETURN
1280 '*****スコープ********
1290 DATA E1,81,81,01,01,01,01,FF
1300 DATA 01,01,01,01,01,81,81,E1
1310 DATA 0E,02,02,00,00,00,00,FE
1320 DATA 00,00,00,00,00,02,02,0E
1330 '*****UF0**********
1340 DATA 00,00,00,01,03,07,3F,7F
1350 DATA D2,7F,3F,07,03,00,00,00
1360 DATA 00,00,00,80,C0,E0,FC,FE
1370 DATA 4B,FE,FC,E0,C0,00,00,00
1380 '*****バクハツ******
1390 DATA 09,49,25,11,C9,25,93,FF
1400 DATA 03,25,C9,11,25,49,09,00
1410 DATA 20,24,48,10,26,48,80,FF
1420 DATA 80,48,26,10,48,24,20,00
```

●ホントにできたかな

最後まで入れたらちょっと実験してみましょう。このままでは何もでないので、100 PUT SPRITE Ø, (1ØØ,1ØØ), 9, 1:GOTO1ØØ を追加して R U N ↵ でプログラムを実行すると、画面のまん中に赤いUFOが出現するはずです。止めるには、CTRL ↗ STOP を押しましょう。

実験が終わったら 1 0 0 ↵ と押して、行番号1ØØを消して次に進みます。



DATA E1, 81, 81, ……これなあに?

これは、16進数といってキャラクターを数字にかえたものなんです。

16進数というのは16になるとØになって次の位に1つあがる数え方です。

1Ø進法	1	2	………	9	1Ø	11	12	13	14	15	16
16進法	1	2	………	9	A	B	C	D	E	F	1Ø

たとえば、計算のしかたは、
1Ø →16 (1×16+Ø)
E1 →225 (14×16+1)
FF →255 (15×16+15)

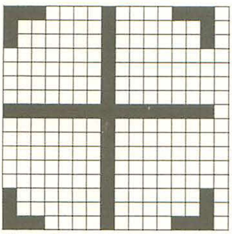
DATAなどのときによく使うからおぼえておいてね。

●キャラクタを作るには

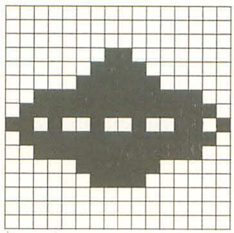
UFOを作ったりするのはスプライト機能を使います。キャラクタはスプライトパターンといって、8×8か16×16の点の組み合せで作られているのです。できあがったスプライトパターンはスプライトプレーンという面にかかれます。

SPRITE$（スプライトダラー）	スプライトパターン(キャラクタ)を登録する
書き方	SPRITE$（スプライトパターン番号）＝データ
使い方	SPRITE$（Ø）＝B$

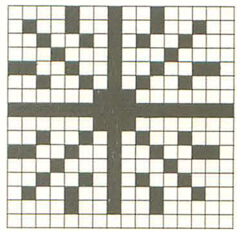
スプライトパターンは8×8ならØ〜255個、16×16ならØ〜63個まで作れるの。1つ1つのスプライト(スプライトパターンをこれからこう呼びます)に番号をつけて、キャラクタを16進の数にして登録します。このゲームでは下の3つのキャラクタが登場です。



（スプライトパターン Ø）



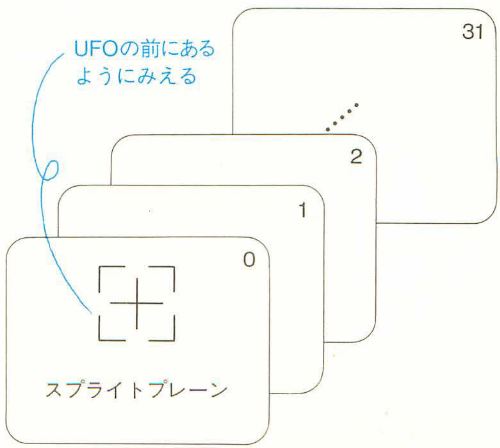
（スプライトパターン 1）



（スプライトパターン 2）

PUT SPRITE（プット スプライト）	スプライト(キャラクタ)を画面にだす
書き方	PUT SPRITE　スプライトプレーン番号,（X,Y),カラー,スプライトパターン番号
使い方	PUT SPRITE　1,（1ØØ,1ØØ),9,1

スプライトプレーンはØから31まであって1つのプレーンに1つのスプライトがだせます。プレーンには優先順位というのがあって重なったときに番号が小さい方が前にくるの。(X, Y)でスプライトをだす位置を決めます。スプライト番号はSPRITE$で登録した番号ね。でる位置が変わるとスプライトが動いているようにみえるんです。



UFO発進

ゲームをつくろう③　スプライトを動かす

次はいよいよ UFO が飛びまわります。今度は NEW はしないで、さっきのプログラムにつけたしていきましょう。

［　　　］の部分を追加します。追加するときは、ただ、[3][0][スペース][O][N]…[↵]と、その行をいれていけばいいの。

```
10 '*****ﾒｲﾝ ﾙｰﾁﾝ******
30 ON STRIG GOSUB600,600 'ﾐｻｲﾙ
40 SCREEN 2,2
50 COLOR 15,1,1 :CLS
70 GOSUB 1200 'ｽﾌﾟﾗｲﾄ
140 X=100:Y=100:T=1000:SC=0
```

```
200 '*****ｳｺﾞｶｽ**********
210 ON SPRITE GOSUB 700 'ｱﾀﾘ
220 INTERVAL ON
230 X1=INT(RND(1)*30)*5+25:Y1=INT(RND(1)
*30)*5+25
240 FOR K1=1 TO 30
250 SPRITE OFF:STRIG(1) ON
260 A=STICK(1)
270 IF (A=2 OR A=3 OR A=4)AND X<180 THEN
 X=X+5
280 IF (A=6 OR A=7 OR A=8)AND X>5 THEN X
=X-5
290 IF (A=1 OR A=2 OR A=8)AND Y>5 THEN Y
=Y-5
300 IF (A=4 OR A=5 OR A=6)AND Y<170 THEN
 Y=Y+5
310 CO=K1 MOD 15+1
320 PUT SPRITE 0,(X,Y),15,0
330 SPRITE OFF
340 PUT SPRITE 1,(X1,Y1),CO,1
350 NEXT K1:GOTO 210
```

```
600 '*****ﾐｻｲﾙ*********
610 STRIG(1) OFF:SPRITE STOP
620 PUT SPRITE 0,(X,Y),15,2
640 SPRITE ON
650 RETURN
```

```
700 '*****アタリ**********
710 SPRITE OFF:K1=30
820 PUT SPRITE 0,(X,Y),15,0
830 RETURN
```

●ちゃんと動くかな

プログラムをいれたら、またまた RUN ↵ してみましょう UFOがあっちこっちに出現したかな？まん中にでているのはスコープです。ジョイパッドを動かしたい方向に押すと、スコープは自由に動きまわります。UFOにねらいをつけて トリガ1 を押すと、レーザー発射！レーザーを発射するとバクハツの絵がパッと現われます。止めるには、やっぱり CTRL ↗ STOP です。

●どうしてレーザーは発射できるの

ON STRIG GOSUB（オン スティックトリガ ゴーサブ）	トリガキー を押したら飛ぶところを決める
STRIG ON/OFF（スティックトリガ オン オフ）	トリガキー の割り込みスイッチ
書き方	ON STRIG GOSUB 行番号 STRIG(番号) ON, STRIG(番号) OFF
使い方	行番号30, 250, 610を見てね

レーザー発射の秘密はON STRIG GOSUBとSTRIG ON/OFFという命令にあったのです。まず、行番号30で"トリガキー を押したら、行番号600に飛ぶ"と決めます。行番号250で"トリガキー を押したら飛ぶ"というスイッチをONにします。スイッチONを忘れたら、いくら飛ぶところを決めてあってもだめですよ。トリガキー を押すと行番号600に飛んで"レーザーを発射する"という仕事をします。レーザーを発射している間はSTRIG OFFでスイッチを切っておきます。レーザーを発射したら、行番号650のRETURNでGOSUBあった次の行にもどります。

何も押さなければ行番号210から350をずっとくりかえします

●レーザーが命中したら

せっかくレーザー砲を発射しても〝UFOにあたったらどうなる〞というのがないとつまりません。この〝あたったらどうなる〞の動きをしてるのがON SPRITE GOSUBとSPRITE ON/OFFです。

ON SPRITE GOSUB（オン スプライト ゴーサブ）	スプライトがぶつかったら飛ぶところを決める
SPRITE ON/OFF（スプライト オン オフ）	スプライトの割り込みスイッチ
書き方	ON SPRITE GOSUB 行番号 SPRITE ON, SPRITE OFF
使い方	行番号21Ø, 64Ø, 71Øを見てね

まず行番号21Øで、UFOにレーザー砲があったとき行番号7ØØに飛ぶように決めます。

下の1コマと2コマ目まではSTRIG ON/OFFを使っています。

レーザー砲を発射したら行番号64ØでSPRITE ONにします。これで、もしバクハツの絵とUFOの絵が同じ位置にあればあたったことになり、行番号7ØØに飛びます。

行番号7ØØからはミサイルがUFOにあたったときにどうするかです。あたったらどうするかは、キミたちが好きなように作れるのです。UFOにあたった時得点がふえたり、音がでたりしたらおもしろそう。

1 ミサイル発射



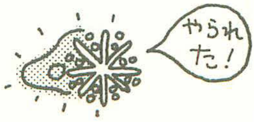
＊＊＊ミサイル＊＊＊に飛ぶ

2 バクハツ



＊＊＊ミサイル＊＊＊のところ

3 命中

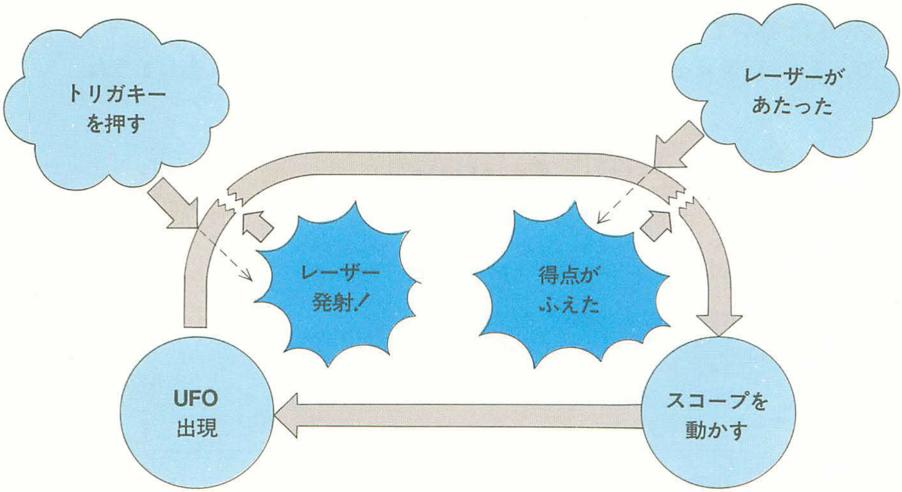


＊＊＊アタリ＊＊＊に飛ぶ

●なくてはならない割り込みの話

STRIG ON/OFF や SPRITE ON/OFF などの命令を使うと、〝トリガキー を押す〃と〝レーザーを発射する〃という動きや〝レーザーがあたる〃と〝得点がふえる〃という動きがゲームの流れに割りこんでくるのです。

このような、〝何かしたら～する〃という働きを「割りこみ」といいます。

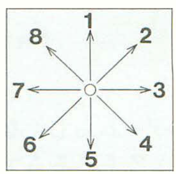


●ところでスコープを動かしているのは

ジョイパッドを押すとスコープが動きましたね。これは、STICK という命令を使っています。

STICK（スティック）	ジョイパッドなどをどの方向に押しているか調べる
書き方	STICK（番号）
使い方	行番号 26Ø を見てね

ゲームでは A＝STICK(1) と使っています。これはポケット A に、どの方向に押したかという数が入ります。方向によって数は右のようになっています。押さないときは Ø ね。(　)の中の番号は何でキャラクタを動かすかによって違いますよ。



STICK(Ø)…	↑ ↓ ← →（カーソルキー）を使うとき
STICK(1)…	コントローラー1につなげたジョイスティックやジョイパッドを使うとき
STICK(2)…	コントローラー2につなげたジョイスティックを使うとき

UFOは宇宙へ

ゲームを作ろう④ 背景を描く

まっ黒の画面をUFOが飛んでるだけというのはちょっとつまらない。そこで、バックの絵を描いてみようではありませんか。

さっそく、行番号を追加します。あとは、行番号400からをいれましょうね。

```
10 '*****ﾒｲﾝ ﾙｰﾁﾝ******
30 ON STRIG GOSUB600,600 'ﾐｻｲﾙ
40 SCREEN 2,2
50 COLOR 15,1,1 :CLS
60 OPEN "GRP:" FOR OUTPUT AS #1
70 GOSUB 1200 'ｽﾌﾟﾗｲﾄ
80 GOSUB 400 'ﾊｲｹｲ
100 IF INKEY$<>CHR$(13) THEN 100
140 X=100:Y=100:T=1000:SC=0
150 LINE(208,100)-(248,118),1,BF
```

```
400 '*****ﾊｲｹｲ*********
410 LINE(0,0)-(201,191),15,B
420 LINE (255,0)-(201,191),15,B
430 CIRCLE(0,191),100,5
440 PAINT(0,191),5
450 RESTORE
460 FOR S=10 TO 170 STEP 10
470 READ S$
480 PRESET(208,S)
490 PRINT #1,S$
500 NEXT
510 RETURN
520 DATA SCORE," 0","",HI-,SCORE," 0",""
,TIME,1000,"","","","",HIT,RETURN,TO,STA
RT
```

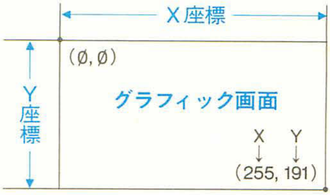
●ゲームらしくなったかな

もうそろそろプログラムをいれるのも速くなったかな？さて、RUNしてみましょう。宇宙の絵と右側にスコアもでてくるはず。「あれれ、UFOとスコープは？」と心配しているキミたち、安心を。[↵]を押してみましょう。UFOとスコープが現れます。

止めるには、[CTRL]↗[STOP]。もう覚えた？

●背景を描くグラフィック命令

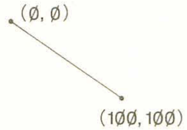
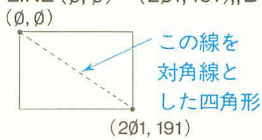
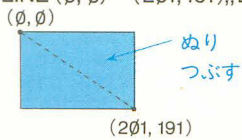
宇宙の絵を描いているのは、ここで紹介する3つのグラフィック命令を使っています。



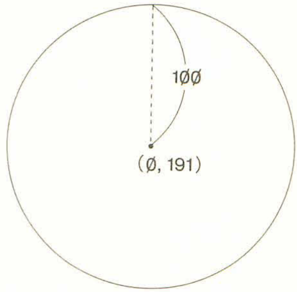
グラフィック命令では左から何ドット目で上から何ドット目かという位置の指定すればいいの。

ライン LINE	線を引く
書き方	LINE (始めの位置)−(終わりの位置), カラー, BまたはBF
使い方	LINE (Ø, Ø)−(2Ø1, 191), 15, B

ワク線を引いているのがLINEです。LINEは次の3つの使い方があります。カラーを決めれば好きな色で線や四角を描けちゃう。

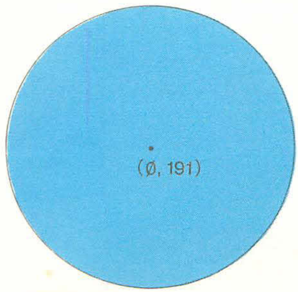
線を引く	四角を描く	四角を描いてぬる
LINE (Ø, Ø)−(1ØØ,1ØØ) 	LINE (Ø, Ø)−(2Ø1, 191),, B 	LINE (Ø, Ø)−(2Ø1, 191),, BF 

サークル CIRCLE	円を描く
書き方	CIRCLE (中心の位置), 半径, カラー
使い方	CIRCLE (Ø, 191), 1ØØ, 5



背景の中で円を描いているのがCIRCLEです。LINEはBFをつければぬりつぶすことができたけどCIRCLEはできません。そこで登場するのがPAINTというぬりつぶす命令です。

ペイント PAINT	ぬりつぶす
書き方	PAINT (位置), カラー, カラー
使い方	PAINT (Ø, 191), 5



円の中で位置を決めれば、その円の中がぬりたい色でぬりつぶされるというわけ。円の中だけじゃなくて、同じ色の線でかこまれた中ならどこでもOKですよ。

UFOバクハツ! ドカーン!

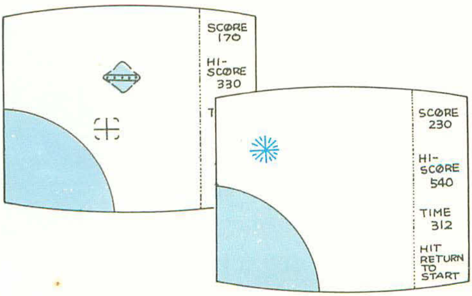
ゲームを作ろう⑤ 音や得点をだす

さあ、最後のしあげ、ゲームオーバーや効果音、ミサイルがあたったときの得点をつけるプログラムをつけたしましょう。

```
10 '*****ﾒｲﾝ ﾙｰﾁﾝ******
20 ON INTERVAL=90 GOSUB 900 'ｼﾞｶﾝ
30 ON STRIG GOSUB600,600 'ﾐｻｲﾙ
40 SCREEN 2,2
50 COLOR 15,1,1 :CLS
60 OPEN "GRP:" FOR OUTPUT AS #1
70 GOSUB 1200 'ｽﾌﾟﾗｲﾄ
80 GOSUB 400 'ﾊｲｹｲ
90 HSC=0
100 IF INKEY$<>CHR$(13) THEN 100
110 LINE(208,20)-(248,28),1,BF
120 PRESET(208,20)
130 PRINT #1," 0"
140 X=100:Y=100:T=1000:SC=0
150 LINE(208,100)-(248,118),1,BF
```

```
200 '*****ｳｺﾞｶｽ**********
210 ON SPRITE GOSUB 700 'ｱﾀﾘ
220 SPRITE OFF:STRIG(1) ON
230 INTERVAL ON
```

```
600 '*****ﾐｻｲﾙ*********
610 STRIG(1) OFF:SPRITE STOP
620 PUT SPRITE 0,(X,Y),15,2
630 BEEP:BEEP
640 SPRITE ON
650 RETURN
```



```
700 '*****ｱﾀﾘ**********
710 SPRITE OFF:K1=30
720 SOUND 7,37:SOUND 6,0
730 SOUND 0,25:SOUND 1,0
740 SOUND 2,25:SOUND 3,0
750 SOUND 8,16:SOUND 9,16
760 SOUND 12,50:SOUND 11,7
770 SOUND 13,0
780 SC=SC+10
790 LINE(208,20)-(248,28),1,BF
800 PRESET(208,20)
810 PRINT #1,SC
820 PUT SPRITE 0,(X,Y),15,0
830 RETURN
```

```
900 '*****ｼﾞｶﾝ**********
910 STRIG(1) OFF:SPRITE OFF
920 LINE(207,88)-(250,98),1,BF
930 T=T-10
940 IF T<0 THEN 960 ELSE PRESET(208,90):
PRINT #1,T
950 RETURN
960 PUT SPRITE 0,(X,208)
970 IF HSC <SC THEN HSC=SC
980 LINE(208,60)-(248,68),1,BF
990 PRESET(208,60)
1000 PRINT #1,HSC
1010 PRESET(208,100)
1020 PRINT #1,"GAME"
1030 PRESET (208,110)
1040 PRINT #1,"OVER"
1050 GOTO 100
```

●ハラハラドキドキいそがないとゲームオーバー

プログラムをいれ終わったらさっそくRUN。さっきと同じように[↵]を押すとゲームが始まります。今度はのんびりしてられません。時間がどんどんなくなります。TIMEが0になったらゲームオーバー。その前にUFOにねらいをつけてどんどん[トリガキー]でレーザーを発射しましょう。レーザーの発射音も出ます。UFOにあたったときとはずれたときは、ちょっと音がちがってます。あたれば、10点。キミたちは何点とれるかな？ゲームオーバーの後、もう1度[↵]を押すとまた始まり。 もし、音がでなかったらテレビのボリュームを上げてみてね。

●ちょっとだけなら BEEP

ミサイルが発射したときにでる音は BEEP という命令を使っています。

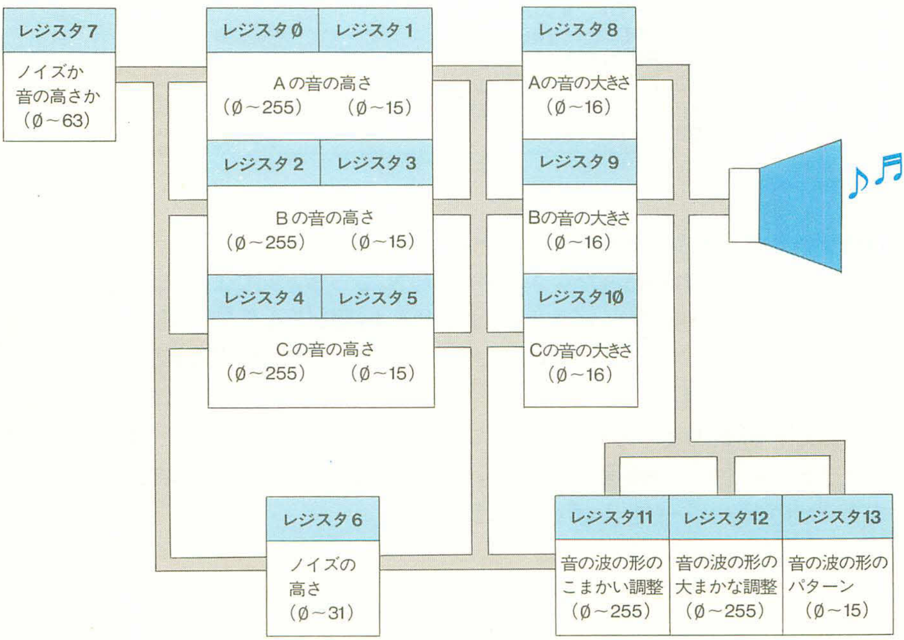
ビープ BEEP	ピッという音をだす
書き方	BEEP
使い方	BEEP 長くだしたいときはBEEP:BEEP:……とつなげてかきます。

●効果音を出す SOUND

BEEPだけではものたりないというとき、SOUNDならいろいろな音をだすことができます。ゲームではレーザーが UFO に当たったときに活躍してます。

サウンド SOUND	効果音をだす
書き方	SOUND レジスタ番号, 数
使い方	SOUND 7, 37

音はチャンネル A、B、C という 3 ヶ所からでています。どんな音をだすか決めるには、0から 13 のレジスタに数字をいれて指定すればいいの。レジスタっていうのは音のポケットみたいなものです。それぞれのレジスタは下のようになっていますよ。レジスタの組み合せをいろいろためして、自分の好きな音をつくってみましょう。



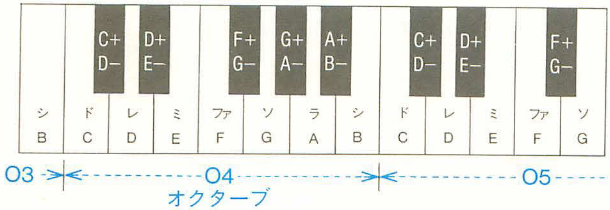
＊( )内は指定する数の範囲

● PLAYで演奏

ゲームの中では使ってないけど、SOUNDやBEEPとともになくてはならないうれしい命令にPLAYがあります。楽器を演奏するのと同じようにPLAYでは〝ドレミファソラシド〟の音階を出すことができちゃう。

プレイ PLAY	ドレミの音をだす
書き方	PLAY〝文字〟
使い方	PLAY〝O4CDEF〟





＊オクターブはO1からO8までであります。

音階は上のようになっています。

PLAY〝O4CDEF〟↵ といれると4オクターブの高さの〝ドレミファ〟の音がでるというわけ。

これだけだと音の長さや速さはいつも同じ。そこで次のように長さをL、速さをT、休符をRで決めると、いろいろに変えられます。



音の長さ	L1（短い）～L64（長い）
音の速さ	T32（おそい）～T256（速い）
休符	R1（休みが短い）～R64（長い）

たとえば上のように作曲したら、こんなふうに書きます。

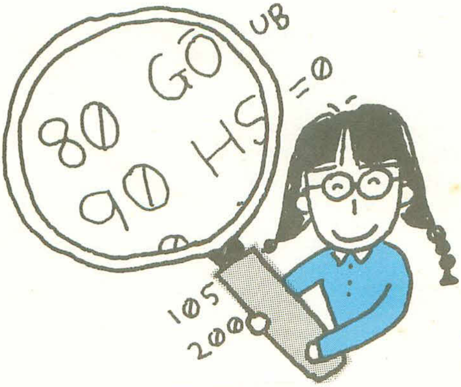
PLAY〝T8Ø L4O4CL8EGL4O5DC〟

ゲーム完成！

```
10 '*****ﾒｲﾝ ﾙｰﾁﾝ******
20 ON INTERVAL=90 GOSUB 900 'ｼﾞｶﾝ
30 ON STRIG GOSUB600,600 'ﾐｻｲﾙ
40 SCREEN 2,2
50 COLOR 15,1,1 :CLS
60 OPEN "GRP:" FOR OUTPUT AS #1
70 GOSUB 1200 'ｽﾌﾟﾗｲﾄ-
80 GOSUB 400 'ﾊｲｹｲ
90 HSC=0
100 IF INKEY$<>CHR$(13) THEN 100
110 LINE(208,20)-(248,28),1,BF
120 PRESET(208,20)
130 PRINT #1," 0"
140 X=100:Y=100:T=1000:SC=0
150 LINE(208,100)-(248,118),1,BF
200 '*****ｳｺﾞｶｽ**********
210 ON SPRITE GOSUB 700 'ｱﾀﾘ
220 INTERVAL ON
230 X1=INT(RND(1)*30)*5+25:Y1=INT(RND(1)
*30)*5+25
240 FOR K1=1 TO 30
250 SPRITE OFF:STRIG(1) ON
260 A=STICK(1)
270 IF (A=2 OR A=3 OR A=4)AND X<180 THEN
 X=X+5
280 IF (A=6 OR A=7 OR A=8)AND X>5 THEN X
=X-5
290 IF (A=1 OR A=2 OR A=8)AND Y>5 THEN Y
=Y-5
300 IF (A=4 OR A=5 OR A=6)AND Y<170 THEN
 Y=Y+5
310 CO=K1 MOD 15+1
320 PUT SPRITE 0,(X,Y),15,0
330 SPRITE OFF
340 PUT SPRITE 1,(X1,Y1),CO,1
350 NEXT K1:GOTO 210
400 '*****ﾊｲｹｲ*********
410 LINE(0,0)-(201,191),15,B
420 LINE (255,0)-(201,191),15,B
430 CIRCLE(0,191),100,5
440 PAINT(0,191),5
450 RESTORE
460 FOR S=10 TO 170 STEP 10
470 READ S$
480 PRESET(208,S)
490 PRINT #1,S$
500 NEXT
510 RETURN
520 DATA SCORE," 0","",HI-,SCORE," 0",""
,TIME,1000,"","","","",HIT,RETURN,TO,STA
RT
```

```
600 '*****ﾐｻｲﾙ********
610 STRIG(1) OFF:SPRITE STOP
620 PUT SPRITE 0,(X,Y),15,2
630 BEEP:BEEP
640 SPRITE ON
650 RETURN
700 '*****ｱﾀﾘ*********
710 SPRITE OFF:K1=30
720 SOUND 7,37:SOUND 6,0
730 SOUND 0,25:SOUND 1,0
740 SOUND 2,25:SOUND 3,0
750 SOUND 8,16:SOUND 9,16
760 SOUND 12,50:SOUND 11,7
770 SOUND 13,0
780 SC=SC+10
790 LINE(208,20)-(248,28),1,BF
800 PRESET(208,20)
810 PRINT #1,SC
820 PUT SPRITE 0,(X,Y),15,0
830 RETURN
900 '*****ｼﾞｶﾝ*********
910 STRIG(1) OFF:SPRITE OFF
920 LINE(207,88)-(250,98),1,BF
930 T=T-10
940 IF T<0 THEN 960 ELSE PRESET(208,90):
PRINT #1,T
950 RETURN
960 PUT SPRITE 0,(X,208)
970 IF HSC <SC THEN HSC=SC
980 LINE(208,60)-(248,68),1,BF
990 PRESET(208,60)
1000 PRINT #1,HSC
1010 PRESET(208,100)
1020 PRINT #1,"GAME"
1030 PRESET (208,110)
1040 PRINT #1,"OVER"
1050 GOTO 100
```

```
1200 '*****ｽﾌﾟﾗｲﾄ*******
1210 RESTORE 1280
1220 FOR P=0 TO 2:B$=""
1230 FOR I=1 TO 32
1240 READ A$
1250 B$=B$+CHR$(VAL("&H"+A$)):NEXT I
1260 SPRITE$(P)=B$:NEXT P
1270 RETURN
1280 '*****ｽｺｰﾌﾟ********
1290 DATA E1,81,81,01,01,01,01,FF
1300 DATA 01,01,01,01,01,81,81,E1
1310 DATA 0E,02,02,00,00,00,00,FE
1320 DATA 00,00,00,00,00,02,02,0E
1330 '*****UF0*********
1340 DATA 00,00,00,01,03,07,3F,7F
1350 DATA D2,7F,3F,07,03,00,00,00
1360 DATA 00,00,00,80,C0,E0,FC,FE
1370 DATA 4B,FE,FC,E0,C0,00,00,00
1380 '*****ﾊﾞｸﾊﾂ******
1390 DATA 09,49,25,11,C9,25,93,FF
1400 DATA 03,25,C9,11,25,49,09,00
1410 DATA 20,24,48,10,26,48,80,FF
1420 DATA 80,48,26,10,48,24,20,00
```



前のページはゲームのプログラムの全部のリストです。もしも、途中で説明のとおりできなかったら、ここでいれまちがいがないかチェックしてみてね。

さて、プログラムの中で使う命令の他の重要な働きをする命令もここでちょっとまとめておきましょう。AUTOやRUNはよくでてきましたよね。

命令	働きと使い方
NEW（ニュー）	PV-7が覚えているプログラムを全部消す。
AUTO（オート）	行番号を自動的にだす。たとえば、 ①AUTO↵ なら行番号10から10ずつふやしてだす。 ②AUTO100↵ なら行番号100から10ずつふやしてだす。 ③AUTO1,1↵ なら行番号1から1ずつふやしてだす。 止めるときは CTRL → STOP
RUN（ラン）	プログラムをRUNする。
LIST（リスト）	プログラムを画面にだす。たとえば、 ①LIST↵ ならプログラムの全部をだす。 ②LIST30↵ なら行番号の30をだす。 ③LIST50-↵ なら行番号50から最後までをだす。 ④LIST-50↵ なら最初から行番号50までをだす。 ⑤LIST10-100↵ なら行番号10から100までをだす。 途中で止めたいときは STOP 、まただしたいときはもう一度 STOP または CTRL → STOP
DELETE（デリート）	決められた行番号を消す。 行番号の決め方はLISTの②④⑤と同じ。 ①と③はできない。
RENUM（リナンバー）	プログラムの行番号をつけ直す。 行番号の決め方はAUTOと同じ。

どんな命令ができたかな

いろいろな命令がでてきましたね。ここでちょっと説明した命令をまとめてみましょう。プログラムリストを見ながらおさらいね。

命　令	働　き
ビープ BEEP	ピッという音をだす
サークル CIRCLE	円を描く
クリアスクリーン CLS	画面の文字などを消す
カラー COLOR	色を決める
ライン LINE	線や四角形を描く
オン　スティックトリガ　ゴーサブ ON STRIG GOSUB スティックトリガ　オン　オフ　ストップ STRIG ON/OFF/STOP	トリガ を押したときの割り込み。STOP では止めておく
オン　スプライト　ゴーサブ ON SPRITE GOSUB スプライト　オン　オフ　ストップ SPRITE ON/OFF/STOP	スプライトがぶつかったときの割り込み。STOPでは止めておく
ペイント PAINT	囲まれた部分をぬる
プレイ PLAY	ドレミの音階をだす
プリント PRINT	文字などを画面にだす
プット　スプライト PUT SPRITE	スプライトを画面にだす
スクリーン SCREEN	どの画面を使うか決める
サウンド SOUND	効果音をだす
スプライトダラ SPRITE$	スプライトを登録する
スティック STICK	ジョイパッドなどの押している方向を調べる。

こんな命令もありました

説明していない命令の他にもまだ、こんな命令を使ってました。☞マークのついている命令はゲーム以外のプログラムでもよくでてくるからあとのページで説明します。☞マーク以外はリファレンス・チャートを見てね。

命令	働き
キャラクターダラ CHR$	キャラクタコードの文字をだす
☞ フォア　ネクスト FOR～NEXT	くりかえす
☞ ゴーサブ　リターン GOSUB～RETURN	決められた行番号へ飛んでまたもどってくる
☞ ゴートゥ GOTO	決められた行番号へ飛ぶ
☞ イフ　ゼン　エルス IF～THEN～ELSE	もし～だったらという条件判断をする
インキーダラ INKEY$	キーボードで押した文字や数字をポケットにいれる
インテジャー INT	少数点を切りすてる
レングス LEN	文字の長さを調べる
☞ レット LET(＝)	ポケットに文字や数字をしまう
オン　インターバル　ゴーサブ ON INTERVAL GOSUB インターバル　オン　オフ　ストップ INTERVAL ON/OFF/STOP	時間の割り込み。ゲームの中ではゲームオーバーになるように使っている
オープン OPEN プリント シャープ PRINT ＃ ポイント リセット PRESET	グラフィック画面に文字や数字をだす。 ゲームの中では右側のスコアの部分の文字などを書いている。
リマークス REM(′)	プログラムの中にメモをいれる
リストア RESTORE ☞ リード READ データ DATA	決められた行番号からデータをポケットにしまう。
ランダム RND	乱数をだす
ライトダラ RIGHT$	文字の右から指定の数の文字をとりだす
ストリングスダラ STR$	数を文字にかえる
バリュー VAL	文字を数にかえる

プログラムの行先表示板



プログラムはRUNすると行番号の小さいものから順に実行しているの。けれど、途中で前の行番号やずっとあとの行番号に飛ぶこともできるの。

飛ぶときには、飛びたいところの行番号を決めればよいのです。

```
200 '*****ウゴカス*************
210 ON SPRITE GOSUB 700 'アタリ
220 INTERVAL ON
230 X1=INT(RND(1)*30)*5+25:Y1
*30)*5+25
240 FOR K1=1 TO 30
250 SPRITE OFF:STRIG(1) ON

320 CO=K1 MOD 15+1
330 PUT SPRITE 0,(X,Y),15,0
340 PUT SPRITE 1,(X1,Y1),CO,1
350 NEXT K1:GOTO 210
```

●どこでも飛んでいくGOTO

GOTO（ゴートゥー）	指定の行番号へ飛ぶ
書き方	GOTO 行番号
使い方	GOTO 21Ø

ゲームのプログラムの中では、行番号33Øででてきて、キャラクターをくりかえしうごかしているんです。

●飛んだらもどってくるGOSUB～RETURN

GOTOは飛んだら行きっぱなしで、またもどれと決めない限りもどってきませんが、GOSUB～RETURNを使うと飛んだ先の仕事がすんだら自動的にもどってくるんです。

GOSUB～RETURN（ゴーサブ リターン）	飛んでまたもどってくる
書き方	GOSUB行番号～RETURN
使い方	ゲームプログラムを見てね

ゲームプログラムの中でGOSUB～RETURNは大活躍をしています。たとえばメインルーチンを見ると行番号7ØでGOSUB 12ØØとしてスプライトをつくるところにとびます。仕事が終わればRETURNでもどってきて、今度は背景を描くところに飛びます。このようにGOSUB～RETURNを使うとプログラムの流れが見やすくなりますよ。

```
10 'メイン ルーチン
20 ON INTERVAL=90
30 ON STRIG GOSUB 6
40 SCREEN 2,2
50 COLOR 15,1,1':CLS
60 OPEN "GRP:" FOR O
70 GOSUB 1200 ' スプラ
80 GOSUB 400 ' ハイケ
90 HSC=0
100 IF INKEY$<>"
```

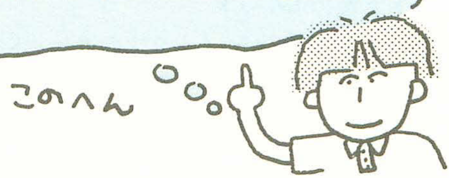
●どっちにしようか判断

GOTO や GOSUB～RETURN は、なにがなんでも飛んでいきます。もし、こうだったらこっちの仕事、そうじゃなかったらこっちの仕事とコンピュータが判断する命令が IF～THEN です。

IF～THEN～ELSE（イフ ゼン エルス）	条件判断をする
書き方	IF 条件 THEN 仕事① ELSE 仕事②
使い方	IF INKEY$<>CHR$(13) THEN 100

条件にあっていれば①の仕事、あっていなかったら②の仕事をして次の行にいきます。仕事②のところに行番号を書けば、その行に飛びます。ELSE から後を書かないと、すぐ次の行にいくんです。ゲームのプログラムの中で IF～THEN は、ジョイパッドを押している方向を判断してスコープを動かしています。

```
70 IF (A=2 OR A=3 OR A=4)AND X<180 THEN
 X=X+5
280 IF (A=6 OR A=7 OR A=8)AND X>5 THEN X
=X-5
290 IF (A=1 OR A=2 OR A=8)AND Y>5 THEN Y
=Y-5
300 IF (A=4 OR A=5 OR A=6)AND Y<170 THEN
 Y=Y+5
```



条件のところでは、こんな記号や文字を使います。

記号や文字	意　味	たとえば
□＝△	□が△と同じ	A=2
□＜＞△	□が△と同じでない	INKEY$<>CHR$(13)
□＜△	□が△より小さい	X<180
□＞△	□が△より大きい	X>5
□＜＝△(□＝＜△)	□が△と同じか△より小さい	B<=3
□＞＝△(□＝＞△)	□が△と同じか△より大きい	B>=3
NOT 条件	条件がちがうなら	NOT A=2
条件① AND 条件②	条件①で条件②なら	A=2 AND A=3
条件① OR 条件②	条件①か条件②なら	A=2 OR A=3

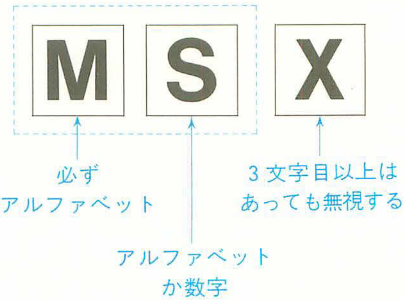
ポケットの話

●ポケットってなに?

今までときどきポケットという言葉が説明の中に出てきましたよね。〝このポケットってなんだ?〟と思っていたキミたちも多いと思います。ポケットっていうのは、プログラムの中で使う文字や数をしまっておくことのできるところなんです。ポケットに文字や数をしまうと PV-7 が、その文字や数を覚えてくれます。

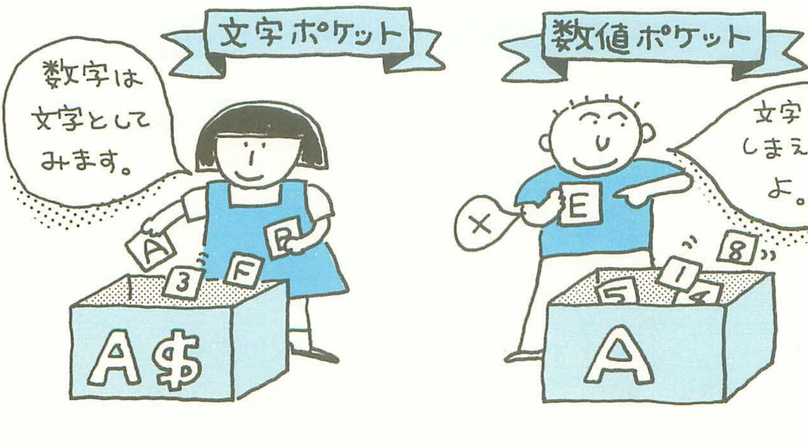
●ポケットに名前をつける

ポケットを使うときには、それぞれ1つずつ名前をつけます。名前は 1～2 文字でつけます。



●文字のポケットと数のポケット

ポケット名のおしりに〝$〟がつくと文字のポケットとして文字をしまいます。〝$〟がついていないと数のポケットとして数値をしまいます。数値っていうのは計算に使う数字のこと。たとえば、電話番号などは、数字でも計算しないから数値ではありません。



●ポケットにデータ(文字や数値)をしまう

LET(レット)	ポケットにデータをしまう
書き方	LET ポケット名＝データ
使い方	LET X＝100, LET A$＝"バーネット"



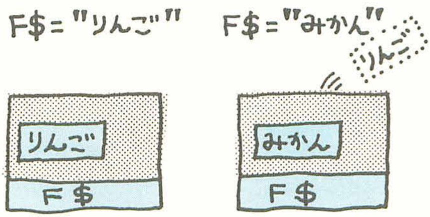
ポケットにデータをしまうのはLETという命令です。PRINTのときと同じように文字をしまうときには"(ダブルクォーテーション)で囲みます。

ところで、ゲームのプログラムを思いだしてください。よく、＝(イコール)という記号がでてきましたね。これは、だいたいLETを省略した形でポケットにデータをしまってたということなのです。

では、もうちょっとポケットについてお話ししましょう。

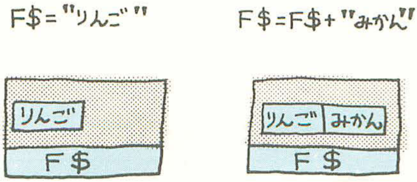
●前のデータは消えてしまう

ポケットF$に"りんご"というデータがはいっているとしましょう。ところが、このF$に"みかん"というデータをいれると"りんご"は消えてしまうのです。



●つけたすことならできる

またまたF$には"りんご"が入っています。これに＋(プラス)を使えば、データをつなげていっしょにいれることができるのです。



●計算した答えをポケットにしまう

計算した答えをポケットにしまうこともできます。ゲームのプログラムの中でSC＝SC＋1Ø なんて使ってましたよね。ポケット SC には得点が入っていて、ミサイルが当たったら 1Ø 点ずつたしていくという働きをしていたのです。

計算するときには、記号にちょっと気をつけてね。コンピュータでは、かけ算とわり算は＊、／を使います。たとえば8×4は8＊4、32÷4は32／4と書きます。



●〝　〞で囲むと意味が変わってしまう

PRINT 命令のところでででてきた〝(ダブルクォーテーション)でポケットを囲むと F＄ という文字になってしまうんです。だから、F＄ の中味を PRINT でだしたいときは〝はつけないでね！

PRINT F$ なら	PRINT "F$"
りんご OK □　　ポケットF$の中味をだす	F$ OK □　　F$という文字をだす



BASIC 便利記号

BASICには便利な特殊記号がいくつかあるので紹介しましょう。

記号	使い方
.（ピリオド）	プログラムをRUNしたらSyntax errorになってしまったとき、 [L][I][S][T][.] とするとその行がでてくる
ー（マイナス）	いくつからいくつまでというときの〝から〞の役目。たとえば、LIST 1ØØ-25Ø
:（コロン）	1行に命令をつなげて書くことができる。たとえば SCREEN 2,2:CLS

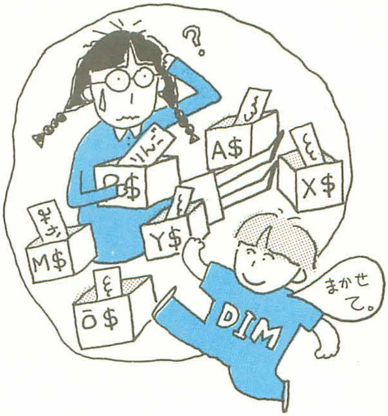
●ポケットをたくさん使うときは

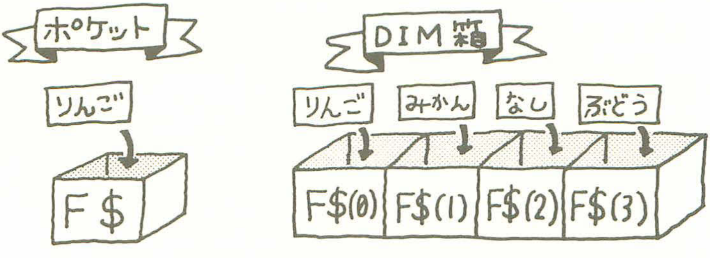
プログラムをつくっていくと、たくさんのポケットを使わなければならないことがあります。あんまりたくさんあると、どのポケットにどんなデータが入っているのかわからなくなったり、なかなかめんどうです。

そこで、ひとつ便利なポケットを紹介しましょう。それは、DIM箱といって、同じ性質のデータをたくさん使うときにとっても便利。

DIM（デイム）	ポケットがたくさんある箱を作る
書き方	DIM 名前(ポケットの数)
使い方	DIM F$(3)

DIM箱を使うときは、こんなDIM箱をいくつ使うよと決めておきます。それがDIMという命令。たとえば、DIM F$(3)とすると、F$(Ø)からF$(3)までの4つのDIM箱が使えるようになるの。





記号	使い方
,(カンマ)	区切りに使う。たとえば、 INPUT A,B,C
;(セミコロン)	区切りに使う。たとえば、 PRINT A$;B$
′(アポストロフィ)	プログラムの中にメモを入れる。たとえば、 ′スプライト
?(クエスチョンマーク)	PRINTの代わりに使う。たとえば、 ? "MSX"

ポケットにデータをしまういろいろ

ここで、またプログラムをいれてみます。NEWを実行するとせっかくのゲームのプログラムが消えてしまうので行番号2Ø1Øから下のプログラムをいれていきます。

```
2010 CLS:INPUT"なまえ";N$
2020 LOCATE 10,10
2030 PRINT N$
2040 FOR Y=15 TO 20
2050 LOCATE 5,Y
2060 FOR I=1 TO4
2070 PRINT"ぱち" SPC(2);
2080 NEXT I
2090 NEXT Y
```

RUN2Ø1Ø↵と押すと行番号2Ø1ØからRUNします。

まず、"なまえ"とでてきます。これはコンピュータが何かをいれるのを待っているのです。キーボードからたとえばバ゛ーネット↵と名前をいれます。PV-7がおばさんに"ぱちぱち"と拍手してくれますよ。

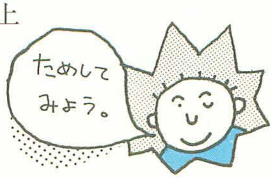
●キーボードからデータをいれる INPUT

このプログラムでは、"バーネット"というデータをポケットN$にしまっています。このように、INPUTという命令を使ってキーボードからいれたデータをポケットにしまうこともできるのです。

INPUT（インプット）	キーボードからデータをいれてポケットにしまう
書き方	INPUT "だしたい文字"; ポケット名, ポケット名
使い方	INPUT "なまえ"; N$

だしたい文字を省略すると?だけ画面にでます。また2つ以上のポケットにデータをいっぺんにいれることもできます。

1	2
INPUT A, B, C ?□　└コンマで区切る	? 3, 4, 5 OK└データもコンマで区切る □



● INPUT を使ったらエラーがでちゃった

INPUTを使ってでるエラーは、こんな意味ですよ。

? Redo from start	??	? Extra ignored
ポケットとデータの型がちがうよ。数のポケットなのに文字をいれたりしてないかな？	いれたデータの数が足りない。データをポケットの数だけいれてね。	いれたデータが多い。余分にいれたデータは無視しますよ。

●好きなだけくりかえすFOR～NEXT

ところで、プログラムでは行番号2070で"ぱち"と1回しか買いていないのに画面には"ぱち"がいくつもでてきましたね。これは、FOR～NEXTを使って何回もくりかえしているからなのです。

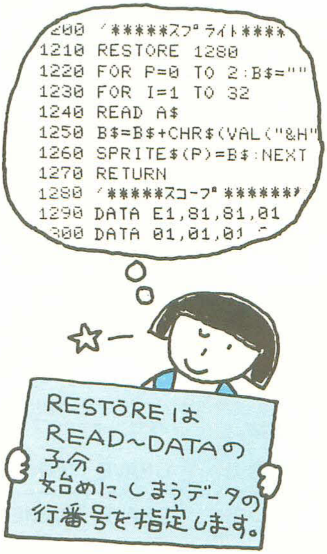
フォー ネクスト FOR～NEXT	何回もくりかえす
書き方	FOR　ポケット名＝始めの数 TO 終わりの数 STEP 増える数 ∫ NEXT ポケット名
使い方	FOR Y＝15 TO 20 ∫ NEXT　Y

プログラムでは行番号2070を4回、さらに2050から2080を6回ずつくりかえしています。

●プログラムの中にデータをかいておく

INPUTはキーボードからデータをいれてポケットにしまったけど、プログラムの中にデータを書いておくこともできます。

ゲームのスプライトづくりのところを思い出してみて。ここで使っていたREAD～DATAがそうです。FOR～NEXTも活躍しているから見てね。



リード データ READ～DATA	データをポケットにしまう
書き方	READ ポケット名 ∫ DATA　データ
使い方	行番号 1240～1290

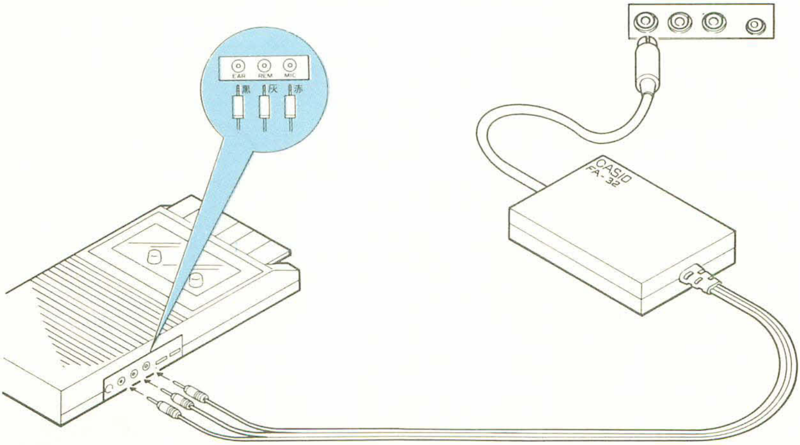
READの後のポケットに、DATAの後のデータを1つずつしまっていきます。READ～DATAを使ってエラーがでたらこんな意味。

Syntax error	Out of DATA	
READのポケットと DATA　のデータの型がちがうよ。	ポケットの数がデータの数より多いよ。	データの数がポケットの数より多いとき、余分なデータは、だまって無視。

プログラムをしまっておこう

さて、せっかくゲームなどのプログラムをいれても、このままでは、電源を切ったらPV-7はプログラムを全部忘れてしまいます。でも、キミたちにカセットインターフェースとカセットテープレコーダーがあればだいじょうぶ。カセットテープに録音しておくことができます。プログラムをとっておくことを〝C SAVE〟（セーブ）といいます。

カセットインターフェースを使ってPV-7とカセットテープレコーダーをつなげたら、カセットテープをセットしましょう。



●プログラムをセーブする

シーセーブ CSAVE	カセットテープにプログラムをセーブする
書き方	CSAVE〝プログラムの名前〟
使い方	CSAVE〝GAME〟



プログラムの名前は6文字以内のアルファベットで好きな名前をつけます。セーブの手順は次のとおりね。

①カセットテープレコーダーの録音ボタンを押す。
②キーボードから命令をいれる。たとえば、C S A V E 〝 G A M E 〟 ↵
③画面にOKがでたらおしまい。
④カセットテープレコーダーのストップを押して止める。

●うまくセーブができたかな

セーブが正しくできていなくて、それを知らずに電源を切ってしまったら、せっかくのプログラムも水の泡。そこで、セーブが正しくできたか調べるのが「CLOAD?」

シーロード? CLOAD?	正しくセーブしたか調べる
書き方／使い方	CLOAD?

カセットテープを巻もどしたら C L O A D ? ↵ といれます。カセットテープレコーダーの再生ボタンを押して少し待ちます。

正しくセーブしてあれば"OK"、そうでなければ"Verify error"とでます。"OK"がでなかっら、もう一度セーブしなおしましょう。

●プログラムをロードする

セーブと反対にカセットテープから PV-7 にプログラムをもってくるのがCLOAD。(ロードするだけで、いれなおさなくてもいいのです。)

ロードの手順は次のとおりです。

シーロード CLOAD	PV-7 にプログラムをロードする
書き方	CLOAD "セーブしたときの名前"
使い方	CLOAD "GAME"

ロードのときの画面



①ロードしたいプログラムの入っているカセットテープをセーブしてあるあたりに巻もどす。
②命令をいれる。たとえば C L O A D " G A M E " ↵。
③"OK"がでたらおしまい。
④カセットテープレコーダーのストップボタンを押す。あとは、R U N ↵ でロードしたプログラムを RUN できるというわけです。

これだけ知っていれば、プログラムづくりもラクですね。おばさんも、もっとおもしろいプログラムづくりにちょうせんしますから、キミたちもがんばってね。

リファレンスチャート

※〔　〕の中は省略できます。

BASICの命令をABC順にならべて説明してあります。プログラムをつくる時に役立ててね。

ASC	文字のキャラクタコードをだす

書き方　ASC　(文字式)
使い方　PRINT ASC　("A")

AUTO	行番号を自動的にだす

書き方　AUTO　始めの行番号[, いくつずつだすか]
使い方　AUTO　2ØØ

BEEP	ピッという音をだす

書き方／使い方　BEEP

BLOAD	機械語プログラムをロードする

書き方　BLOAD　"デバイス名[ファイル名]"[, R][, オフセット]
使い方　BLOAD　"CAS:MSX", R

BSAVE	機械語プログラムをセーブする

書き方　BSAVE　"デバイス名[ファイル名]", 開始アドレス, 終了アドレス[, 実行アドレス]
使い方　BSAVE　"CAS:MSX", &H5Ø, &HA2FF

CALL	拡張命令を呼びだす

書き方　CALL　拡張命令[引数]
使い方　CALL TALK　("A")

CHR $	キャラクターコードの文字をだす

書き方　CHR $　(キャラクターコード)
使い方　CHR $　(33)

CIRCLE	円を描く

書き方　CIRCLE　(中心の位置), 半径[, カラー]
使い方　CIRCLE　(Ø, 191), 1ØØ, 5

CLEAR	ポケットをからにする、メモリーとして使うところの大きさを決める

書き方　CLEAR　[文字領域の大きさ][, メモリ領域の大きさ]
使い方　CLEAR　4ØØ

CLOAD　　パソコンのメモリにプログラムをロードする

書き方　CLOAD　[〝セーブしたときの名前″]
使い方　CLOAD　〝GAME″

CLOAD?　　正しくセーブしたか調べる

書き方　CLOAD?　[〝ファイル名″]
使い方　CLOAD?

CLOSE　　ファイルを閉じる

書き方　CLOSE　[#][ファイル番号][, ファイル番号]…
使い方　CLOSE　#1

CLS　　画面の文字や絵を消す

書き方／使い方　CLS

COLOR　　色を決める

書き方　COLOR　[文字の色][, 背景の色][, 周辺の色]
使い方　COLOR　15, 6, 3

CONT　　一度止めたプログラムをまたつづける

書き方／使い方　CONT

CSAVE　　PV-7からカセットテープにプログラムをセーブする

書き方　CSAVE　〝プログラムの名前″[, ボーレート]
使い方　CSAVE　〝GAME″

DEF FN　　自分で関数を作る

書き方　DEF FN　名前[(引数)]＝作った命令
使い方　DEF FNB　(X, Y)＝X＊2

DEFINT/DEFSNG　　ポケットの型を決める。それぞれ, 整数型,
DEFDBL/DEFSTR　　単精度実数型, 倍精度実数型, 文字型にする

書き方　DEFINT　文字の範囲　(他のも同じ)
使い方　DEFINT　A-C　(他のも同じ)

DELETE　　決められた行番号を消す

書き方　DELETE　[行番号][－終わりの行番号]
使い方　DELETE　1Ø　または、DELETE　5Ø－1ØØ

DIM　ポケットがたくさんある箱を作る

書き方　DIM　名前(ポケットの数−1)
使い方　DIM　F$(3)

DRAW　画面に図形を描く

書き方　DRAW　"グラフィックの命令1つ1つの意味"
使い方　DRAW　"BM 100, 100 D40R40U40L40"

END　プログラムを終わりにする

書き方／使い方　END

ERASE　DIMで作ったポケットを消す

書き方　ERASE　DIM箱の名前[, DIM箱の名前…]
使い方　ERASE　F$

ERROR　エラーをだす

書き方　ERROR　エラーコード
使い方　ERROR　200

FOR〜NEXT　FOR文からNEXTまでの命令を決められた回数だけくりかえす

書き方　FOR　ポケット名＝始めの数TO終わりの数[STEP増える数]
　　　　NEXT　ポケット名[, ポケット名……]
使い方　FOR　P＝0　TO 2
　　　　NEXT　P

GOSUB〜RETURN　指定したサブルーチンへ飛び、実行したらもどってくる

書き方　GOSUB　行番号
　　　　RETURN
使い方　GOSUB　1200
　　　　RETURN

GOTO　指定の行番号へ飛ぶ

書き方　GOTO　行番号
使い方　GOTO　200

IF〜THEN〜ELSE　条件判断をする

書き方　IF　条件　THEN　文または行番号　ELSE　文または行番号
使い方　IF　INKEY$＜　＞CHR$(13)　THEN　100

INKEY$　押しているキーの文字をポケットにいれる　押していなければ空白をだす

書き方　INKEY$
使い方　A$＝INKEY$

INPUT　キーボードからデータをいれてポケットにしまう

書き方　INPUT　[〝だしたい文字〃;]ポケット名[, ポケット名…]
使い方　INPUT　〝CASIO〃; N$

INSTR　文字列の中から決めた文字列の位置さがして、その位置を答えにする

書き方　INSTR　([さがし始める位置,]文字列1, 文字列2)
使い方　PRINT INSTR　(〝D〃, 〝ABCD〃)

INT　小数点以下を切りすてた整数を求める

書き方　INT　(数式)
使い方　PRINT INT　(2.34)

INTERVAL ON/OFF/STOP　時間の割り込みスイッチを ON, OFF, または保留する

書き方／使い方　INTERVAL ON/OFF/STOP

KEY　ファンクションキーの内容を決める

書き方　KEY　キー番号, 書きたい内容
使い方　KEY　1, 〝PRINT〃

KEY LIST　ファンクションキーのリストをだす

書き方／使い方　KEY LIST

KEY (n) ON/OFF/STOP　ファンクションキーの割り込みスイッチの ON,OFF,または保留

書き方　KEY　(キー番号)ON/OFF/STOP
使い方　KEY　(1)ON

KEY ON/OFF　ファンクションキーの内容を画面の下に表示する／しないを判断する

書き方／使い方　KEY ON/OFF

LEFT$　文字列の左側から決めた長さの文字列をとりだす

書き方　LEFT$　(文字, 数式)
使い方　PRINT LEFT$　(〝ABCD〃, 2)

LEN	文字列の全部の文字数を調べる

書き方　LEN　(文字)
使い方　PRINT LEN　("ABCD")

LET	ポケットにデータをしまう

書き方　[LET] ポケット名＝データ
使い方　X＝1ØØ

LINE	直線または四角形を描く

書き方　LINE　(始めの位置)－(終わりの位置)[, カラー][, BまたはBF]
使い方　LINE　(Ø, Ø)－(2Ø1, 191), 15, B

LINE INPUT	キーボードからデータを区切らずいれて，ポケットにしまう

書き方　LINE　INPUT["表示したい文字";]文字のポケット名
使い方　LINE　INPUT"TEL";T$

LIST	プログラムを画面にだす

書き方　LIST　[始めの行番号][－終わりの行番号]
使い方　LIST または LIST　3Ø－1ØØ

LLIST	プログラムをプリンタから打ちだす

書き方　LLIST　[行番号][－行番号]
使い方　LLIST　5Ø－1ØØ　行番号の指定の仕方はLISTと同じ

LOAD	アスキー形式のプログラムをロードする

書き方　LOAD　"デバイス名[ファイル名]", [R]
使い方　LOAD　"CAS:GAME"

LOCATE	テキスト画面のカーソル位置を指定する

書き方　LOCATE　[左から何文字目], [上から何行目], [カーソルスイッチ]
使い方　LOCATE　1Ø, 5

LPRINT	文字などのデータをプリンタから打ちだす

書き方　LPRINT　[打ちだしたいデータ][, または;打ちだしたいデータ…]
使い方　LPRINT　A, B

LPRINT USING	データをプリンタで決められた書き方で打ちだす

書き方　LPRINT USING　書式;打ちだしたい文字列や数値
使い方　LPRINT USING "￥￥＃＃＃";123　書き方はPRINT USINGと同じ

MAXFILES　同時に使用するファイルの数を決める

書き方　MAXFILES＝ファイルの数
使い方　MAXFILES＝4

MERGE　2つのプログラムをつなげる

書き方　MERGE　〝デバイス名[ファイル名]〞
使い方　MERGE　〝CAS : DEMO〞

MOTOR ON/OFF　カセットテープレコーダーのモーターのON/OFF

書き方／使い方　MOTOR　[ON/OFF]

MID＄　文字のポケットの一部を決めた文字に置きかえる

書き方　MID＄　(ポケット名, 何番目から[, 何個])＝置きかえるもの
使い方　MID＄　(A＄, 2, 3)＝〝ABC〞

NEW　本体メモリーが覚えているプログラムを全部消す

書き方／使い方　NEW

ON ERROR GOTO　エラーがおこったときに飛ぶ行を決める

書き方　ON ERROR GOTO　行番号
使い方　ON ERROR GOTO　1ØØ

ON～GOSUB　決めた行番号へ飛んでもどる

書き方　ON式GOSUB　行番号
使い方　ON A GOSUB　1ØØ, 2ØØ, 3ØØ

ON～GOTO　決めた行番号へ飛ぶ

書き方　ON式GOTO　行番号
使い方　ON A GOTO　1ØØ, 2ØØ, 3ØØ

ON INTERVAL GOSUB　時間の割り込みをできるようにする

書き方　ON INTERVAL＝時間GOSUB行番号
使い方　ON INTERVAL＝9Ø GOSUB 9ØØ

ON KEY GOSUB　ファンクションキーによる割込みができるようにする

書き方　ON KEY GOSUB　[行番号][, 行番号……]
使い方　ON KEY GOSUB　1ØØ, 2ØØ

ON SPRITE GOSUB　スプライトがぶつかったときに飛ぶところを決める

書き方　ON SPRITE GOSUB　行番号
使い方　ON SPRITE GOSUB　7ØØ

ON STOP GOSUB　CTRL + STOP の割込みができるようにする

書き方　ON STOP GOSUB　行番号
使い方　ON STOP GOSUB　1ØØ

ON STRIG GOSUB　トリガキーを押した時飛ぶところを決める

書き方　ON STRIG GOSUB　行番号
使い方　ON STRIG GOSUB　6ØØ

OPEN　ファイルをオープンする

書き方　OPEN　〝デバイス名[ファイル名]〞[FORモード]AS#ファイル番号
使い方　OPEN　〝GRP:〞FOR OUTPUT AS #1

PAINT　わくでかこまれた中をぬりつぶす

書き方　PAINT　(X, Y)[, ぬりたい色][, わくの色]
使い方　PAINT (Ø, 195), 5

PEEK　メモリーの決められた番地の内容をよみだす

書き方　PEEK　(番地)
使い方　A＝PEEK　(&HFF)

PLAY　ドレミの音をだす

書き方　PLAY　〝音をだすための命令の1つ1つ〞
使い方　PLAY　〝O4CDEF〞

POKE　メモリ上の決めた番地へデータを書きこむ

書き方　POKE　番地, データ
使い方　POKE　& HFØØØ, & HFF

PRESET　決めた位置の点を消す

書き方　PRESET　(座標)[, カラー]
使い方　PRESET　(2Ø8,6Ø)

PRINT　画面に文字や数字や記号をだす

書き方　PRINT　〝だしたいもの〞
使い方　PRINT　〝カシオ〞

PRINT USING　決められた書き方で文字列や数値を画面にだす

書き方　PRINT USING　書式;書きたいもの
使い方　PRINT USING　"¥ ¥###";123

PRINT#　ファイルにデータを書く

書き方　PRINT#　ファイル番号, 書きたいもの
使い方　PRINT#　1, "SCORE"

PSET　決めた位置に点を打つ

書き方　PSET　(X, Y)[, カラー]
使い方　PSET　(5Ø,1ØØ), 8

PUT SPRITE　スプライト(キャラクタ)を画面にだす

書き方　PUT SPRITE　スプライトプレーン番号[, (X, Y)][, カラー][, スプライトパターン番号]
使い方　PUT SPRITE　1, (1ØØ,1ØØ), 9, 1

READ〜DATA　データをポケットにしまう

書き方　READ　ポケット名〜DATA　データ
使い方　READ　A$〜DATA　E1, 81, 81

REM　プログラムの中にメモを書く

書き方　REM　書きたいメモ
使い方　' メインルーチン　(' はREMの省略形)

RENUM　プログラムの行番号をつけ直す

書き方　RENUM　[新始めの行番号][, 旧始めの行番号][, 増分]
使い方　RENUM　1ØØ, 5Ø

RESUME　エラー処理を終わらせて, プログラムをまた実行する

書き方　RESUME　[再開するところ]
使い方　RESUM　NEXT

RIGHT$　文字の右から決めた長さの文字をだす

書き方　RIGHT$　(文字, 数式)
使い方　PRINT RIGHT$　("ABC", 2)

RND　Ø以上1未満の乱数をだす

書き方　RND　(数式)
使い方　PRINT　RND(1)

RUN　プログラムを RUN する

書き方／使い方　RUN

SCREEN　画面を切りかえる

書き方　SCREEN　画面, スプライトの大きさ
使い方　SCREEN　2, 2

SOUND　効果音をだす

書き方　SOUND　レジスタ番号, 数
使い方　SOUND　7, 37

SPC　決めた数の空白をだす

書き方　SPC　(数式)
使い方　PRINT　"PV";SPC(5);"7"

SPRITE ON/OFF/STOP　スプライトの割込みスイッチ

書き方／使い方　SPRITE ON/OFF/STOP

SPRITE$　スプライトパターン(キャラクタ)を登録する

書き方　SPRITE$　(スプライトパターン番号)＝データ
使い方　SPRITE$　(Ø)＝B$

STICK　ジョイパッド(ジョイスティック)などをどの方向に押しているか調べる

書き方　STICK　(番号)
使い方　A＝STICK　(1)

STOP　プログラムの実行を止める

書き方／使い方　STOP

STOP ON/OFF/STOP　CTRL ↗ STOP の割り込みスイッチ

書き方／使い方　STOP ON/OFF/STOP

STRING$　決めた文字を決めた数だけだす

書き方　STRING$　(だしたい数, だすもの)
使い方　STRING$　(1Ø, "＊")

STR$　数値を文字にする

書き方　STR$　(数式)
使い方　A$＝STR$　(426)

STRIG ON/OFF/STOP　　トリガキーの割り込みスイッチ

書き方　　STRIG(番号)　ON/OFF/STOP
使い方　　STRIG(Ø)　ON

SWAP　　2つのポケットの中身を交換する

書き方　　SWAP　変数名, 変数名
使い方　　SWAP　A$, B$

TAB　　文字を消しながら決められた位置までカーソルを飛ばす

書き方　　TAB　(数式)
使い方　　PRINT　TAB(5)"BASIC"

TRON　　プログラムのどこを実行しているかだす

書き方／使い方　　TRON

TROFF　　プログラムのどこを実行しているかをだすのをやめる

書き方／使い方　　TROFF

VAL　　文字を数値にする

書き方　　VAL　(文字式)
使い方　　A=VAL　("34")

VPEEK　　VRAM上の決められた番地の内容をよみだす

書き方　　VPEEK　(番地)
使い方　　B%=VPEEK(2)

VPOKE　　VRAMの決めた番地にデータを書き込む

書き方　　VPOKE　番地, データ
使い方　　VPOKE　1%, &HFF

WAIT　　プログラムの実行を中断する

書き方　　WAIT　ポート番号, 式1[, 式2]
使い方　　WAIT　1, &H22, &H22

WIDTH　　画面に表示する文字数を指定する

書き方　　WIDTH　文字数
使い方　　WIDTH　2Ø

エラーメッセージ

エラー番号	メッセージ	メッセージの意味
56	Bad file name	ファイル名がまちがっています。
52	Bad file number	ファイル番号がちがっています。
17	Can't CONTINUE	もう続けられません。
19	Device I/O error	カセットやプリンタに入出力エラーがおきました。
57	Direct statement in file	アスキーファイルロード中にプログラムに使えない命令がありました。
11	Division by zero	ゼロで割ってはいけません。
54	File already open	オープンしてあるファイルをもう1度オープンしようとしました。
53	File not found	指定したファイルはありません。
59	File not OPEN	ファイルをオープンしていません。
12	Illegal direct	この命令はプログラムの中でしか使えません。
5	Illegal function call	命令の使い方がちがってます。
55	Input past end	最後のデータの読み込みが終わっているのにさらに読もうとしています。
51	Internal error	メモリやテキストの内容の状態が正しくありません。(電源を入れなおして下さい。)
25	Line buffer overflow	入れた文字数が多すぎます。
24	Missing operand	命令をつたえるのに必要なデータがたりません。
1	NEXT without FOR	NEXT文にたいするFOR文がありません。
21	NO RESUME	ON ERRORがあるのにRESUMEがありません。
4	Out of DATA	データがたりません。

エラー番号	メッセージ	メッセージの意味
7	Out of memory	これ以上プログラムは入りません。
14	Out of string space	文字ポケットの場所がたりなくなりました。
6	Overflow	計算の結果や入力された数値が 大きすぎてあつかえません。
10	Redimensioned array	同じ名前の DIM 箱を作ろうとしています。
22	RESUME without error	エラーがおこっていないのに RESUME を 実行しました。
3	RETURN without GOSUB	RETURN 文と GOSUB 文が 正しく対応していない。
58	Seguential I/O only	シーケンシャルファイルに対してランダムファイルの命令をいれました。
16	String formula too complex	文字の式が複雑すぎます。
15	String too long	文字が長すぎます。
9	Subscript out of range	DIM で決めた数より多いポケットを 使おうとしています。
2	Syntax error	文の書き方がちがってます。
13	Type mismatch	数のポケットに文字をしまおうとしました。
8	Undefined line number	ない行番号を指定しています。
18	Undefined user function	DEF FN で決めていない命令を使おうと しました。
23	Unprintable error	エラーメッセージにはないエラーがおこり ました。
20	Verify error	正しくセーブができていません。

キャラクター・コード表

キャラクターコードというのは、キーボードにある文字や記号をコンピュータにおぼえさせるために数字に直してあるものです。プログラムの中で、CHR$(145)と使うと、これは"あ"になります。PRINT命令以外で文字や数字をだしたいときにはこの表をみてください。

コード(10進)	コード(16進)	CHR	コード(10進)	コード(16進)	CHR	コード(10進)	コード(16進)	CHR	コード(10進)	コード(16進)	CHR	コード(10進)	コード(16進)	CHR	コード(10進)	コード(16進)	CHR
0	00	↑	43	2B	+	86	56	V	129	81	♥	172	AC	ャ	215	D7	ラ
1	01	│	44	2C	,	87	57	W	130	82	♣	173	AD	ュ	216	D8	リ
2	02	│	45	2D	−	88	58	X	131	83	◆	174	AE	ョ	217	D9	ル
3	03	│	46	2E	.	89	59	Y	132	84	○	175	AF	ッ	218	DA	レ
4	04	│	47	2F	/	90	5A	Z	133	85	●	176	B0	ー	219	DB	ロ
5	05	│	48	30	0	91	5B	[	134	86	を	177	B1	ア	220	DC	ワ
6	06	│	49	31	1	92	5C	¥	135	87	ぁ	178	B2	イ	221	DD	ン
7	07	│	50	32	2	93	5D	]	136	88	ぃ	179	B3	ウ	222	DE	゛
8	08	│	51	33	3	94	5E	∧	137	89	ぅ	180	B4	エ	223	DF	゜
9	09	コ	52	34	4	95	5F	—	138	8A	ぇ	181	B5	オ	224	E0	た
10	0A	ン	53	35	5	96	60	'	139	8B	ぉ	182	B6	カ	225	E1	ち
11	0B	ト	54	36	6	97	61	a	140	8C	ゃ	183	B7	キ	226	E2	つ
12	0C	ロ	55	37	7	98	62	b	141	8D	ゅ	184	B8	ク	227	E3	て
13	0D	ー	56	38	8	99	63	c	142	8E	ょ	185	B9	ケ	228	E4	と
14	0E	ル	57	39	9	100	64	d	143	8F	っ	186	BA	コ	229	E5	な
15	0F	キ	58	3A	:	101	65	e	144	90		187	BB	サ	230	E6	に
16	10	ャ	59	3B	;	102	66	f	145	91	あ	188	BC	シ	231	E7	ぬ
17	11	ラ	60	3C	<	103	67	g	146	92	い	189	BD	ス	232	E8	ね
18	12	ク	61	3D	=	104	68	h	147	93	う	190	BE	セ	233	E9	の
19	13	タ	62	3E	>	105	69	i	148	94	え	191	BF	ソ	234	EA	は
20	14	│	63	3F	?	106	6A	j	149	95	お	192	C0	タ	235	EB	ひ
21	15	│	64	40	@	107	6B	k	150	96	か	193	C1	チ	236	EC	ふ
22	16	│	65	41	A	108	6C	l	151	97	き	194	C2	ツ	237	ED	へ
23	17	│	66	42	B	109	6D	m	152	98	く	195	C3	テ	238	EE	ほ
24	18	│	67	43	C	110	6E	n	153	99	け	196	C4	ト	239	EF	ま
25	19	│	68	44	D	111	6F	o	154	9A	こ	197	C5	ナ	240	F0	み
26	1A	│	69	45	E	112	70	p	155	9B	さ	198	C6	ニ	241	F1	む
27	1B	│	70	46	F	113	71	q	156	9C	し	199	C7	ヌ	242	F2	め
28	1C	│	71	47	G	114	72	r	157	9D	す	200	C8	ネ	243	F3	も
29	1D	│	72	48	H	115	73	s	158	9E	せ	201	C9	ノ	244	F4	や
30	1E	│	73	49	I	116	74	t	159	9F	そ	202	CA	ハ	245	F5	ゆ
31	1F	↓	74	4A	J	117	75	u	160	A0		203	CB	ヒ	246	F6	よ
32	20		75	4B	K	118	76	v	161	A1	。	204	CC	フ	247	F7	ら
33	21	!	76	4C	L	119	77	w	162	A2	「	205	CD	ヘ	248	F8	り
34	22	"	77	4D	M	120	78	x	163	A3	」	206	CE	ホ	249	F9	る
35	23	#	78	4E	N	121	79	y	164	A4	、	207	CF	マ	250	FA	れ
36	24	$	79	4F	O	122	7A	z	165	A5	・	208	D0	ミ	251	FB	ろ
37	25	%	80	50	P	123	7B	{	166	A6	ヲ	209	D1	ム	252	FC	わ
38	26	&	81	51	Q	124	7C	|	167	A7	ァ	210	D2	メ	253	FD	ん
39	27	'	82	52	R	125	7D	}	168	A8	ィ	211	D3	モ	254	FE	
40	28	(	83	53	S	126	7E	~	169	A9	ゥ	212	D4	ヤ	255	FF	
41	29	)	84	54	T	127	7F		170	AA	ェ	213	D5	ユ			
42	2A	*	85	55	U	128	80	♠	171	AB	ォ	214	D6	ヨ			

グラフィックキャラクタコード表

10進数のコード0～31(16進数では00～1F)は制御文字が割り当てられています。
この範囲の文字の表示は PRINT CHR$(1)＋CHR$(表示文字コード＋64)と行ないます。
月の表示→PRINT CHR$(1)＋CHR$(65)(月の表示文字コードは1)

コード(10進)	コード(16進)	キャラクタ	コード(10進)	コード(16進)	キャラクタ	コード(10進)	コード(16進)	キャラクタ	コード(10進)	コード(16進)	キャラクタ	コード(10進)	コード(16進)	キャラクタ	コード(10進)	コード(16進)	キャラクタ
0	00		6	06	土	12	0C	秒	18	12	┬	24	18	┌	30	1E	中
1	01	月	7	07	日	13	0D	百	19	13	┤	25	19	┐	31	1F	小
2	02	火	8	08	年	14	0E	千	20	14	├	26	1A	└			
3	03	水	9	09	円	15	0F	万	21	15	┼	27	1B	┘			
4	04	木	10	0A	時	16	10	π	22	16	│	28	1C	╳			
5	05	金	11	0B	分	17	11	円	23	17	─	29	1D	大			

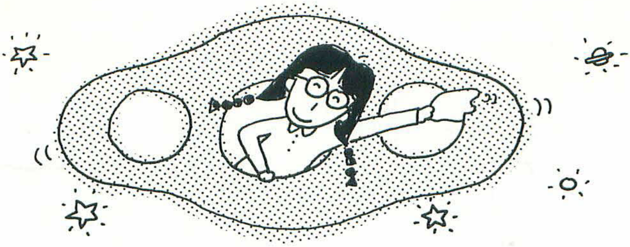
さくいん

あいうえお順

アルファベット順

BASICガイドは、これでおしまい。
後で"こういうときにはどうするんだっけ？"というときにまた、
BASICガイドを開いてみてください。

キミたちが、面白いゲームを作ってくれることを
おばさんは期待しています。



それじゃあ、またね。

PV-7 BASICガイド

昭和59年 8 月25日　初版発行
昭和59年10月15日　第 3 版発行
発行 —— カシオ計算機株式会社
〒160　東京都新宿区西新宿2-6-1
新宿住友ビル
企画・編集 —— カシオ計算機株式会社
宣伝企画部
制作 —— SOMARS

CASIO®

SM0300-01483C